NPS-GSBPP-09-013

NAVAL
POSTGRADUATE
SCHOOL

MONTEREY, CALIFORNIA

SHARE Repository Component Specification: Needs Assessment

19 August 2009

by

Jean Johnson, Research Assistant

Graduate School of Engineering & Applied Sciences

Naval Postgraduate School

Prepared for: Naval Postgraduate School, Monterey, California 93943

Naval Postgraduate School
Monterey, California

Daniel T. Oliver
President

Leonard A. Ferrari
Provost

The Acquisition Chair, Graduate School of Business & Public Policy, Naval Postgraduate School supported the funding of the research presented herein.

The report was prepared by:

_____
Jean Johnson, Research Assistant
Graduate School of Engineering & Applied Sciences

Reviewed by:

_____
William R. Gates, Ph.D.
Dean, Graduate School of Business & Public Policy

Released by:

_____
Karl van Bibber, Ph.D.
Vice President and
Dean of Research

REPORT DOCUMENTATION PAGE

Form approved
OMB No 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	19 August 2009	1 October 2007 through 30 September 2008

4. TITLE AND SUBTITLE
SHARE Repository Component Specification: Needs Assessment

5. FUNDING
N/A

6. AUTHOR (S)
Jean Johnson

7. PERFORMING ORGANIZATION NAME (S) AND ADDRESS (ES)
NAVAL POSTGRADUATE SCHOOL
GRADUATE SCHOOL OF BUSINESS AND PUBLIC POLICY
555 DYER ROAD
MONTEREY, CA 93943-5103

8. PERFORMING ORGANIZATION REPORT NUMBER
NPS-GSBPP-09-013

9. SPONSORING/MONITORING AGENCY NAME (S) AND ADDRESS (ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words.)

In August 2006, PEO IWS established the Software, Hardware Asset Reuse Enterprise (SHARE) repository to make combat system software and related assets available to eligible current and potential Navy contractors. PEO IWS is seeking ways to improve and mature the capability provided by SHARE. To that end, a research project at the Naval Postgraduate School will produce a component specification and ontology framework for use in SHARE. The framework will expand the information contained in the current metadata, to enable improved search and discovery capabilities and facilitate use of the repository items once they are retrieved. This paper lays the foundation for the research, by providing a characterization of the problem domain by describing the SHARE repository, its contents and its unique attributes. Based on this investigation, we then provide specific recommendations for both near term and long term improvements. The near term suggestions are essentially "low hanging fruit", or ideas for quick improvements that can be implemented in a relatively short time frame. The long term improvements are associated with the implementation of the component specification and ontology. Finally, we outline the requirements for the component specification in terms of its intended use within SHARE.

14. SUBJECT TERMS
Software, Hardware Asset Reuse Enterprise (SHARE) repository

15. NUMBER OF PAGES
47

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT: UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE: UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT: UNCLASSIFIED	20. LIMITATION OF ABSTRACT: UU

NSN 7540-01-280-5800

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Abstract

In August 2006, PEO IWS established the Software, Hardware Asset Reuse Enterprise (SHARE) repository to make combat system software and related assets available to eligible current and potential Navy contractors. PEO IWS is seeking ways to improve and mature the capability provided by SHARE. To that end, a research project at the Naval Postgraduate School will produce a component specification and ontology framework for use in SHARE. The framework will expand the information contained in the current metadata, to enable improved search and discovery capabilities and facilitate use of the repository items once they are retrieved. This paper lays the foundation for the research, by providing a characterization of the problem domain by describing the SHARE repository, its contents and its unique attributes. Based on this investigation, we then provide specific recommendations for both near term and long term improvements. The near term suggestions are essentially “low hanging fruit”, or ideas for quick improvements that can be implemented in a relatively short time frame. The long term improvements are associated with the implementation of the component specification and ontology. Finally, we outline the requirements for the component specification in terms of its intended use within SHARE.

THIS PAGE INTENTIONALLY LEFT BLANK

Acknowledgements

The author would like to acknowledge Dr. Mikhail Auguston for his contributions to the vision for the component specification and ontology. She would also like to thank the sponsor, Mr. Nick Guertin, PEO IWS 7B, the SHARE program manager, Ms. Melody Belcher, and Mr. Mark Wessman for their guidance throughout this research.

THIS PAGE INTENTIONALLY LEFT BLANK

About the Author

Jean Johnson is a research assistant in the Naval Postgraduate School Systems Engineering Department. She coordinates a program to develop Modeling and Simulation curriculum for DoD Acquisition workforce personnel and performs research for the PEO IWS SHARE program. Previously, Ms. Johnson held various positions supporting NAVSEA's Warfare Systems Engineering Division. Jean is a U.S. Navy active and reserve veteran and continues her Navy affiliation in the Individual Ready Reserve. She holds a M.E. in Operations Research and Systems Analysis and a B.S. in Applied Mathematics from Old Dominion University and is currently pursuing her PhD in Software Engineering at NPS.

Jean Johnson
Graduate School of Engineering and Applied Sciences
Naval Postgraduate School
777 Dyer Road, Room 209
Monterey, CA 93943-5000
Tel: 831-656-2956
Fax: (831) 656-3129
E-mail: jmjohnso@nps.edu

THIS PAGE INTENTIONALLY LEFT BLANK

NPS-GSBPP-09-013

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

SHARE Repository Component Specification: Needs Assessment

19 August 2009

by

Jean Johnson, Research Assistant

Graduate School of Engineering & Applied Sciences

Naval Postgraduate School

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

THIS PAGE INTENTIONALLY LEFT BLANK

List of Acronyms

IDS	Interface Design Specification
LCS	Littoral Combat System
NESI	Net-centric Enterprise Solutions for Interoperability
NPS	Naval Postgraduate School
PEO IWS	Program Executive Officer, Integrated Warfare Systems
PEO C4I	Program Executive Officer, Command, Control, Communications, Computers and Intelligence
PIDS	Prime Item Development Specification
ReSEARCH	Requirements Search Engine
SSDS	Ship Self Defense System
SRS	Software Requirements Specifications
SSS	System/Subsystem Specifications
SHARE	Software, Hardware Asset Reuse Enterprise
TSCEI	Total Ship Computing Environmental Infrastructure
UML	Unified Modeling Language
XML	Extensible Markup Language

THIS PAGE INTENTIONALLY LEFT BLANK

Introduction

The purpose of this paper is to lay the foundation for the Naval Postgraduate School SHARE component specification and ontology research project funded by Program Executive Officer, Integrated Warfare Systems (PEO IWS). It is intended as a communication forum between the stakeholders and project performers to ensure congruence of goals and to validate requirements. First, we characterize the problem domain by describing the Software, Hardware Asset Reuse Enterprise (SHARE) repository, its contents and its unique attributes. Based on this investigation, we then provide specific recommendations for both near-term and long-term improvements. The near-term suggestions are essentially "low hanging fruit," or ideas for quick improvements that can be implemented in a relatively short time frame. The long-term improvements are associated with the benefits that can be realized once the component specification and ontology have been implemented. Finally, we outline requirements for the component specification in terms of its intended use within SHARE.

THIS PAGE INTENTIONALLY LEFT BLANK

Background

In August 2006, PEO IWS established the SHARE repository to make available combat system software and related assets to current and potential Navy contractors (PEO IWS Library, 2007, February 6). SHARE is one piece of the Navy's Open Architecture (OA) approach to developing modular, open systems (PEO IWS, 2007), which includes reusable software applications as a core principle.

PEO IWS is currently seeking ways to improve and mature the capability provided by SHARE. Among other initiatives, two related research projects are in progress at NPS. The first, and the topic of this paper, will produce a component specification framework and ontology for use in SHARE. The component specification is essentially a model of the assets incorporated into the repository; these will enable robust search and discovery capabilities, asset submission assistance, and other repository functions. The ontology is a framework for the relationships between components, providing contextual meaning to asset descriptions. The second project will develop a prototype of a semantically based requirements search engine (ReSEARCH) with the tools necessary to convert documents into semantically based formal representations of requirements (Martel, 2007).

What is SHARE?

SHARE provides a capability for discovering, accessing, sharing, managing, and sustaining reusable assets for the Navy Surface Domain's programs (Belcher, 2007). SHARE consists of an asset library and a card catalog. The asset library is a collection of combat systems software and supporting artifacts. The card catalog is a web-based interface that facilitates user insight into the contents of SHARE and supports user functions—including account registry, asset search and discovery, asset submission assistance, and asset retrieval requests.

The SHARE asset library is separate from the card catalog for two primary reasons. First, the majority of the contents of SHARE is classified material and,

therefore, must be kept in a SECRET or higher container. Second, the process for retrieving assets from SHARE includes necessary steps for addressing the data rights associated with each component. For most of the components, a license agreement and Non-Disclosure Agreement are required before an asset can be issued. Due to these restrictions, the web interface and the actual assets are physically separated.

The search and discovery process in SHARE is conducted either through individual navigation of the list of assets in the catalog (see Appendix B) or by a keyword search of more detailed descriptions. From the catalog list, a user can select an asset for the detailed description, which consists of identity, description and usage information if they are available. The identity information includes asset point of contact, ID, name, version, type, editor and update information. The asset description includes a free-text overview, classification level, export control and distribution statements, current state of the asset, artifact types and usage instructions. Usage information includes user agreement, subscriber, and user information.

The metadata for assets is collected during the asset submission process via an excel spreadsheet available on the SHARE user interface (see Appendix A). Submitters download the spreadsheet and then email the completed form to the SHARE helpdesk. This information includes not only contributor and asset descriptions, but also begins to address domain-specific information by identifying the asset's tie to the generic architecture provided by the Surface Navy OA Warfare Systems Architecture Element Level Decomposition.

Assets are requested from SHARE using an online interactive questionnaire. The user is asked several basic questions, such as which assets are being requested, the justification for asset retrieval, and delivery information. The tool then prepares the necessary documents (including non-disclosure and license agreements) and provides them, along with instructions for printing and submission,

to the user. Once the documents have been mailed in to the SHARE administrators, the user can track the status of the request online through the SHARE interface.

The SHARE user interface also includes some administrative information, such as points of contact for the SHARE program, the list of registered users, a document library, and a calendar. There is also a place where users can post feedback. However, this feature has not yet been utilized.

What is in SHARE?

The contents of SHARE are listed in Appendix B. Currently, SHARE includes the software and supporting documentation for an Aegis Baseline (7.1.1.1), the DDG1000 Total Ship Computing Environmental Infrastructure (TSCEI), and Ship Self Defense System (SSDS) Mk 2 Mod 1. For the Aegis baseline, the source code applications for all major subsystems with build files are included in the repository, as well as Prime Item Development Specifications (PIDS), computer program requirements specifications, Interface Design Specifications (IDS), and user manuals. The TSCEI assets include both documentation and source code. SSDS submissions include the System/Subsystem Specifications (SSS), Software Requirements Specifications, (SRS) and source code for major subsystems. Additionally, the repository includes the Littoral Combat System (LCS) Open Data Model, which provides the mission architecture for LCS (Fein, 2007, February).

What makes SHARE unique?

Several aspects of the SHARE repository make it unique in comparison to any number of existing software repositories—such as SourceForge (2007) or Koders (2007). The first unique attribute is that the current artifacts incorporated in the database are very similar. They are each large subsystems of combat systems for Navy surface platforms. They have a similar level of granularity (very large and complex), and they are all traceable to a subset of the Surface Navy OA Warfare Systems Architecture Element Level Decomposition.

While this observation seems to point to trivial solutions for the repository, consideration of the future of the repository yields a different perspective. A primary realization is that the number of artifacts in the library will continue to grow. At some point, the number of items alone will render the search and discovery process difficult if not aided by visualization tools and robust search engines. Furthermore, if the goal of enterprise-wide, repository-enabled software reuse is to be realized, it is likely that the artifact characteristics will evolve over time. As Open Architecture becomes a standard development approach, more modular systems will be introduced. Once that occurs, it will be advantageous to developers to be able to identify and retrieve modules rather than subsystems. In other words, active repository use is likely to stimulate more granular activity. Additionally, to enable enterprise-level asset sharing, the repository must support the expression of component capability and utility in a meaningful way across domains. It is also important to note that SHARE is intended to include hardware artifacts, although these types of items are not currently included in the card catalog. In summary, it is expected that over time, the artifacts in SHARE will both become more heterogeneous, as well as be required to hold meaning among other more heterogeneous artifacts.

Another unique characteristic of SHARE is that there is no immediate access to assets in the repository. Due to classification and data rights issues, we must distinctly separate the tools used for search and discovery from the components themselves. We cannot insist, for example, that the component specification become part of the component as a wrapper and expect the tools to interface with it directly. These classification and data rights issues compel another important consideration. Since one of the most cumbersome processes identified for SHARE is the navigation of access authority and permissions for component retrieval, solutions aimed towards improving the usability of the repository should incorporate mechanisms for aiding in this process.

An additional distinguishing characteristic of SHARE is the part it plays in the context of the Navy enterprise. Each of the items in SHARE represent “product

lines” in the surface domain, and the surface domain is a part of the larger Navy enterprise. This framework provides contextual meaning to the assets and also becomes the driving force for the desired relevance of tools developed for SHARE. Where possible, it is desirable to incorporate the domain information related to an asset to maximize its contextual meaning. Additionally, as tools are designed, developers should consider their potential use in the larger enterprise domain.

THIS PAGE INTENTIONALLY LEFT BLANK

Recommendations for Near-term Improvement

Throughout this initial research of the SHARE repository, we have identified several relatively uncomplicated improvements. These improvements can be implemented with the repository in its current state before any fundamental framework is put in place. We offer these suggestions for consideration by SHARE leadership to enable near-term enhancement of the capability. These recommendations include improved use of the metadata, increased website functionality, and SHARE education.

The current metadata collected for assets submitted in SHARE includes a free text overview of the asset. These descriptions are currently the best tools that users have to determine if the asset being considered is going to be valuable for them to retrieve. However, the information provided varies greatly in these descriptions. On one end of the spectrum, the descriptions provide an overview of what the component does in the system as well as information to aid in its use. On the other end, very little additional information is provided. In some cases, the acronyms that are listed in the card catalog are simply repeated. Without a better description, users must already know a significant amount about the asset in order to decide if it will be useful to them.

Descriptions should be written with the assumption that the user does not already know what item(s) he/she is seeking. This may be a difficult perspective for program developers to take as they write summaries of their systems. A template could possibly be provided to delineate the types of information required for a description in order to ensure that the appropriate level of detail is included. This description should cover what the component does, its contribution to the overall functionality of a system, and examples of how the component has been used, both in the initial system and as a reused item. Another useful item for searchers less knowledgeable about the various combat systems is an acronym list.

Several features that are popular in commercial search and discovery web interfaces such as Amazon, Google, or Netflix may also be implemented in SHARE to improve the utility of the repository. Customer reviews, frequently asked questions, and tools for visualization are integral sources of information in these websites that could be useful in the SHARE environment as well.

The Amazon model for customer reviews could be beneficial to repository users that have identified an item that looks interesting. Amazon posts the customer ratings, a numeric assignment of quality, and also enables written feedback from the customer. For SHARE, this feedback could be tailored to answer specific questions that users would find useful. Customer feedback would include the quality assessment of the items, a description of how the customer used the component, and lessons learned regarding the item's use. As in Amazon, the SHARE tool could be set up to automatically distribute periodic e-mails requesting customers to review items they have retrieved.

Information visualization aids can help people quickly identify the items of interest to them. A commonly used feature in commercial sites is the "People who bought this, also bought…" feature. This quickly points users to items they may not have been aware of, but which may be relevant in solving their problem. Netflix allows users to view the details about a video in a window that pops up automatically when they move the cursor over a movie cover graphic. This feature may be helpful to those navigating SHARE by allowing users to view the detailed descriptions of components without having to click on them and wait for the information to open. Another improvement that may help provide contextual significance to repository items makes use of the reference architecture information. Currently, the link between the component and the SNOA reference architecture is collected at the time of asset submission. It may be simple to build a search interface based on this mapping. As a search option, the user could choose to display the architecture framework, and then navigate to the components in the repository by clicking on the individual module entities.

A Question and Answer (Q&A) blog could be connected to each of the repository assets. Users interested in an asset would post questions they have about components that seem initially attractive, and asset owners would post answers. Over time, the Frequently Asked Questions (FAQ) can be collected for quick reference. Also, FAQs may reveal a lack of critical information in a component description, which can then be worked into the component metadata. The Q&A blogs themselves may provide valuable information to users, as well. The same concept can also be applied to the SHARE repository overall.

Our final recommended near-term improvement is less a technical solution than it is a cultural solution. One of the reasons that existing examples of reuse are successful is that people understand what they are reusing. We reuse our own code, data structures, and design patterns because we already know them and understand what they can do for us. To that end, education is critical. Before beginning a browse or search, people should understand in general what kind of information is available and how it can be used. This can be presented as a brief write-up (similar to portions of this paper) or as a simple interactive tutorial. Real examples of uses of SHARE would be valuable material to potential SHARE users as well and should be included in the information provided.

THIS PAGE INTENTIONALLY LEFT BLANK

The Long-term Vision

The goal of this research is to improve the development and use of software repositories by developing a component specification designed for use in model-based applications that greatly improve the effectiveness of a software repository. We will develop a specification framework which includes a model of both the components in the repository as well as of the relationships that provide contextual meaning. The component models will be based on the behavior of the component as well as examples of its uses, both within the original system and in any situations in which the component has been reused. The relationships may exist between components within the repository, between the components and a reference or domain architecture, the component's place in the software lifecycle; this and other relational information will aid users in understanding the context of the component.

This framework will enable tools to be developed that will maximize the utility of the reuse repository. Two different types of tools have been identified as necessary for users to make full use of the framework. The search and discovery tools are meant to use the information captured in the framework to assist the user in identifying and retrieving useful items from the repository. In general, it is advantageous for software developers to provide multiple ways for users to search for relevant items; if given such selection, users can investigate the options depending on their background and current needs. To facilitate this process, we envision both advanced visualization tools (such as a fish-eye graph) as well as tools that enable searching from available documentation (such as ReSEARCH). The third type of tool needed is that aimed at assisting component developers by minimizing the overhead associated with creating the component model and inserting it into the repository. One example is a specification-building tool with a wizard-type interface that will assist the developer in creating the component specification. Additionally, a tool is necessary at repository-submission time to help the submitter integrate the component into the repository by building the necessary relationships into the component metadata for proper placement into the repository.

The component specification framework will incorporate all of the information that is collected through the existing efforts to collect SHARE metadata. This includes both the information collected through the current excel sheet, as well as any of the short-term improvements implemented in the interim.

To support the continued and evolutionary use of the specification framework, developers will give consideration throughout its development to potentially changing aspects of SHARE, as well as to including additional candidate repositories. As discussed previously, it is likely that the items placed in SHARE will evolve over time from large subsystems to more granular modules. The component specification should be able to support this evolution of the contents. The framework will also be developed to support multiple repositories. While portions of the framework will contain domain-specific information, the structure and non-domain-specific portions should be easily portable to other repositories. They should also provide a systematic approach to completing the domain-relevant portion. Particular attention will be paid to existing DoD and other software repositories—especially those under the umbrella of the Navy OA domains such as the PEO C4I Net-centric Enterprise Solutions for Interoperability (NESI) repository. The specification framework should also support the integration of these repositories as intended by OA leadership.

Finally, it will be important to integrate the technical solutions provided by this work into the larger effort to improve software reuse within the Navy/DoD. Education, motivation and rewards are needed in order to stimulate the reuse cycle. In addition to the entire domain repository effort, a structured, planned and effective education campaign for these technical solutions is needed.

Requirements for Component Specification

Based on the initial investigation into SHARE as described in previous sections, the requirements listed here for the component specification framework are necessary to providing a solution relevant to the SHARE repository. These items will be considered throughout the framework development.

1. Improved search and discovery capability—The central focus of the specification framework is to facilitate the search and discovery process for a repository. This includes not only the ease of navigation through the available components, but also the completeness of the information. The goal is to educate users about the candidate components thoroughly enough that they know what it is they are retrieving prior to going through that process.

2. Minimize overhead for component submission—The addition of this capability to the repository will come with tradeoffs. Items must conform to the framework in order to be entered into the repository. The time required to prepare an asset for submission into the repository should be minimized as much as possible to avoid disuse due to unacceptable levels of difficulty. The specification framework will support the development of tools to aid the development of the component specification for an asset and to assist integration into the repository.

3. Support multiple user perspectives—The component specification will incorporate multiple views for aiding users in deciding which components to retrieve. These perspectives include, but are not limited to:

    a. Domain-specific reference architecture—Where possible, it is desirable to incorporate into the repository framework the available system domain information related to an asset to maximize its contextual meaning. This may be pre-existing in the form of a reference architecture or some other materials.

    b. Examples of previous uses—Examples of the components' previous uses should be incorporated into the framework. This includes each component's use in the original system as well as any available examples of its reuse in later systems. Both successful and non-successful reuse examples can be included.

    c. Intra-repository component relationships—A visualization of the relationships among the components in the repository is also useful. Items that have been used in the same system or used to perform similar functions in different systems can be grouped together. Additionally, the threads of components that have been reused and reinserted back into the repository as part of a new system should be traceable.

    d. Lifecycle activity information—Information about the lifecycle phase or activity that the artifact is intended to support is useful as well. For example, a user may wish to search for all requirements documentation for systems that perform similar functions to their intended new system as a useful reference.

4. Support for security requirements—Due to the classified nature of the assets in SHARE, the interface for search and retrieval must be kept separate from the assets themselves. Therefore, the specification model must support this constraint of separate locations. Additionally, the metadata of classified elements must be constrained to unclassified material and could possibly include pointers to classified descriptions.

5. Support for legal concerns—As discussed in the previous sections, one of the primary difficulties specific to SHARE is the navigation of access authority and permissions for component retrieval. Any solution provided must take into account these constraints and should incorporate mechanisms to assist this process wherever possible.

6. Extensible to other domains—Since SHARE is part of a greater effort to improve software reuse across the DoD, the component specification framework should support this goal. To that end, the framework should be extensible to the other domains under the Navy OA construct and should support the integration of these capabilities. Additionally, as supporting tools are designed, developers should consider their potential use in the larger enterprise.

7. Scalable for repository evolution—The specification framework should support the evolution of the repository, both from the perspective of the expected growth in the number of components contained, as well as the progression towards less homogenous contents (smaller modules vs. large subsystems, various asset types—design artifacts, documentation, etc.). Additionally, the models should be capable of representing hardware artifacts that may be included as assets in the repository in the future.

8. Use of de facto standards—Wherever possible, implementation of the component specification framework will employ de facto standards such as the Unified Modeling Language (UML), Extensible Markup Language (XML), or others in order to promote broader applicability of existing tools—as well as open an unbiased competition through which tools can be developed.

Future Work

This paper is the first in a series of intermediate products related to the development of the component specification and ontology of SHARE. Future writings will include the results from an ongoing survey of SHARE users and other feedback that has been collected, case studies outlining success and failure stories, and intermediate deliverables supporting the larger task. Near-term research activities will be focused on existing research and practical applications of repository submission procedures, repository management tools, component specification, and model-driven software development (particularly what models are used during various phases of software development) to determine if there are existing solutions that will be relevant in accomplishing the goals of the project.

THIS PAGE INTENTIONALLY LEFT BLANK

List of References

Belcher, M. (2007). PEO IWS software hardware asset reuse enterprise (SHARE). Information brief

Fein, G. (2007, February). Navy's SHARE repository seeing steady growth in first six months. *Defense Daily.*

Koders. (2007). *Open Source Code Source Engine – Koders*. Retrieved October 7, 2007, from www.koders.com.

Martel, C. (2007). *ReSEARCH: A requirements search engine*. Proposal for Future Combat Systems Open Architecture.

PEO IWS. (2007). SHARE home page. Retrieved October 7, 2007, from https://viewnet.nswc.navy.mil/PNB/share/share.nsf/homepage?openform.

PEO IWS Library. (2007, February 6). Software, Hardware Asset Reuse Enterprise (SHARE) Special Notice (DON-SNOTE-070206-005).

SourceForge. (2007). *SourceForge.net: Welcome to SourceForge.net.* Retrieved October 8, 2007, from www.sourceforge.net.

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix A. SHARE Asset Contribution Form

UNCLASSIFIED			
SHARE Asset Contribution v6			
Complete the yellow areas below, and return to: HelpDesk@Nice-Help.net, with cc: to melody.belcher@navy.mil and gregory.hartwig@navy.mil (Items with gray-fill labels will not be published)		SHARE Control Number: (assigned by Help Desk)	
	Asset Name:		
	Asset Description:		
	Request Date:		
Contributor:	Name:		
	Phone:		
	E-Mail ID:		
	Organization:		
	Mailing Address:		
Government Major Program Manager (MPM):	Program Title:		
	Name:		
	Phone:		
	E-Mail ID:		
	Organization + Code:		
	Approval:		
MPM Alternate:	Name:		
	Phone:		
	E-Mail ID:		
	Organization + Code:		
Rationale for Contribution:			
Impacts:			

Asset:	Asset Type:	Sub-Type:	Populate one selection below with a description of the type of asset
	Tactical Application	System	
		Application Program	
		Package	
		System Service	
		Component(s)	
		Library	
		Module/Code Fragment	
		Database / Data Files	
	Development Support	Framework	
		Tools / Utilities	
		Test Tools/ Environments	
	Non-code	Enterprise Framework	
		Data Architecture	
		Pattern / Design / Algorithm	
		Standard / Interface / API	
	New, Modified, or Linked:		
	Dependencies on other assets, COTS, etc.		
	Version:		
	Description:		

Date of Asset:	
Target OS:	
Acquisition or Final?	
Test Level:	
Certification Level:	
OACE Level (self-assessment):	
OAAT Level (self-assessment):	
Complete?	
Buildable?	
Planned Updates:	
Usage Instructions:	

Types of artifacts included within the asset:

		Included? (Y/N)	Format (e.g., DOORS, MS-Word, etc.)
Requirements:	Requirements Specification:		
	Requirements Database:		
Design:	Design Models		
	Design Documents:		
	Patterns:		
	Algorithms:		
	White Papers:		
	Data Models:		
	Simulation Models:		
Code:	Source Code:		
	Compiled Libraries:		
	Executable Programs:		
Test:	Test Plan:		
	Test Procedures:		
	Test Results:		
	Test Tools/Scripts:		
	Test Source Data Files:		
	Test Truth Data:		
	Simulators:		
Interface:	IRSs/IDDs		
	IDSs		
	APIs		
Architecture:	Architecture Model:		
	Architecture Document:		
Supporting Artifacts:	User Documentation:		
	Training Documentation:		
	Build Scripts/Instructions:		
	Other:		

Software Programming language and Operating System(s) Supported

	Pgm Language(s):	
	Run time Environment(s):	
Media Description:	Security Classification:	
	Program's Security Classification Guide ID#:	
	Has MPM pre-approved classification release authority?	

	Media Format:		
	Number of Files:		
	Structure of Files:		
	Total Data Size:		
	Element	Applic.? (Y or blank)	Notes
Architectural Elements (check all that apply):	Middleware/OS		
	Host Application		
	Infrastructure Services		
	Intelligence		
	Track Management		
	Common C2 Services		
	Operational C2		
	Tactical C2		
	Mission Planning		
	Resource Management		
	NTM Tasking/Status		
	Common Display Services		
	Common Operator Displays (e.g., GUIs)		
	Platform Specific Operator Displays		
	Platform Specific Display Devices		
	Local & Offboard Sensor Control		
	Sensor Adaptation		
	Sensor		
	Sensor Stimulation / Simulation		
	Communications Control		
	Communications Adaptation		
	Communications Devices		
	EXCOMM Simulation / Stimulation		
	Off-board Organic Vehicle Control		
	Off-board Organic Vehicle Adaptation		
	Off-board Organic Vehicle		
	Vehicle Simulation / Stimulation		
	Weapon Control		
	Weapon Adaptation		
	Weapon		
	Weapon Simulation / Stimulation		
	Specialized Trainer		
	Ship Control		
	Computing Hardware		
	Engineering / Damage Control		
	Readiness / Support Adaptation		

Microsoft PowerPoint Slide

	Training Control		
	Training Assessment		
	Training Dev. Env.		
	Readiness / Support		
Distribution Statement:			
Data Rights Markings:			
Commercial Software:			
Special Licenses:			
Open Source Software Licenses:			
Data Rights Assertions:			
Any Additional Information:			

Appendix B. SHARE Contents (as of 07 Oct 07)

Name	State	Type	POC	Version
AEGIS				
A-spec: WS-21200/5 SCN 1	Available	Documentation	Andy Li	7.1.1.1
B1-specs: ACTS WS-33417/2	Available	Documentation	Andy Li	7.1.1.1
B1-specs: ADS WS-10666/4	Available	Documentation	Andy Li	7.1.1.1
B1-specs: C&D WS-21208/6	Available	Documentation	Andy Li	7.1.1.1
B1-specs: FCS WS-10521/7	Available	Documentation	Andy Li	7.1.1.1
B1-specs: ORTS WS-10523/10	Available	Documentation	Andy Li	7.1.1.1
B1-specs: SPY WS-10520/10	Available	Documentation	Andy Li	7.1.1.1
B1-specs: WCS WS-10522/9	Available	Documentation	Andy Li	7.1.1.1
B5-specs: TCP WS-33419/2A VOL 1-2	Available	Documentation	Andy Li	7.1.1.1
B5-specs: ADS WS-21366/4A VOL 1-41	Available	Documentation	Andy Li	7.1.1.1
B5-specs: C&D WS-21240/4A VOL 1-28	Available	Documentation	Andy Li	7.1.1.1
B5-specs: FCS WS-10557/12A	Available	Documentation	Andy Li	7.1.1.1
B5-specs: ORTS WS-21234/6A	Available	Documentation	Andy Li	7.1.1.1
B5-specs: SPY WS-10554/16A VOL 1-3	Available	Documentation	Andy Li	7.1.1.1
B5-specs: WCS WS-10555/17A VOL 1-6	Available	Documentation	Andy Li	7.1.1.1
IDS-specs: NAV/AWS S9427-AN-IDS-020/WSN-7	Available	Documentation	Andy Li	7.1.1.1 (31 July 1997)
IDS-specs: WCS/SPY WS-19632/10A	Available	Documentation	Andy Li	7.1.1.1
IDS-specs: SPY/SPY SIG PRO WS-19634/8A	Available	Documentation	Andy Li	7.1.1.1
IDS-specs: FCS/FCS DCC WS-19640/4A	Available	Documentation	Andy Li	7.1.1.1
IDS-specs: ORTS/WCS WS-19644/10A VOL 1-2	Available	Documentation	Andy Li	7.1.1.1
IDS-specs: ORTS/SPY 19646/12A	Available	Documentation	Andy Li	7.1.1.1
IDS-specs: WCS/LAMPS WS-19657/1	Available	Documentation	Andy Li	7.1.1.1. (01 Mar 2000)
IDS-specs: ACTS/SPY WS-19681/8A	Available	Documentation	Andy Li	7.1.1.1
IDS-specs: ACTS/WCS WS-19682/10A	Available	Documentation	Andy Li	7.1.1.1
IDS-specs: ADS/ORTS WS-21267/2A VOL 1-2	Available	Documentation	Andy Li	7.1.1.1
IDS-specs: ADS/C&D WS-21272/2A VOL 1-2	Available	Documentation	Andy Li	7.1.1.1
IDS-specs: ORTS/ACTS WS-21278/2A	Available	Documentation	Andy Li	7.1.1.1
IDS-specs: ADS/ACTS WS-21286/2A	Available	Documentation	Andy Li	7.1.1.1
IDS-specs: ORTS/SCA WS-21287/1A	Available	Documentation	Andy Li	7.1.1.1
IDS-specs: ACEG/AP WS-21288A PT 1-5	Available	Documentation	Andy Li	7.1.1.1

IDS-specs: AP/AOCD WS-21290/1A	Available	Documentation	Andy Li	7.1.1.1
IDS-specs: SPY/C&D WS-21327/8A	Available	Documentation	Andy Li	7.1.1.1
IDS-specs: C&D/WCS WS-21328/7A VOL 1-2	Available	Documentation	Andy Li	7.1.1.1
IDS-specs: ORTS/C&D WS-21329/6A	Available	Documentation	Andy Li	7.1.1.1
IDS-specs: ACTS/C&D 21338/7A VOL 1-2	Available	Documentation	Andy Li	7.1.1.1
S/W: Aegis C&D source code	Available	Application	Andy Li	7.1.1.1
S/W Aegis FCS source code	Available	Application	Andy Li	7.1.1.1
S/W Aegis WCS source code	Available	Application	Andy Li	7.1.1.1
S/W Aegis SPY source code	Available	Application	Andy Li	7.1.1.1
Aegis Quick Reference Guides (QRGs)	Available	Documentation	Andy Li	7.1.1.1
Aegis Interface Design Specifications (IDSs)	Available	Documentation	Andy Li	7.1.1.1
Aegis Interface Design Specs/ ACD-9072_3	Available	Documentation	Andy Li	7.1.1.1
Aegis Interface Design Specs/ WS-10512-2A	Available	Documentation	Andy Li	7.1.1.1
Aegis Reusable Components (ARC) User Manuals	Available	Documentation	Andy Li	7.1.1.1
S/W: Aegis C&D build/support files	Available	Code	Andy Li	7.1.1.1
S/W: Aegis Reusable Components (ARC)	Available	System Service	Andy Li	7.1.1.1
DDG 1000				
TSCEI 4.1 Documentation	Acquisition	Documentation	Tom Kostyo	4.1
TSCEI 4.1 Source Code	Acquisition	Application	Tom Kostyo	4.1
TSCEI 4.2.2 Documentation	Acquisition	Documentation	Tom Kostyo	4.2
LCS				
LCS Data Model 2006-11-22	Acquisition	Architecture/Design	Belcher_MelodyS	11/22/2006
LCS Open Data Model Package—5/22/2007	Acquisition	Architecture/Design	NA	3/20/2007
SSDS				
SSS: SSDS MK 2 System/Subsystem Specification	Available	Documentation	Andy Li	MK 2 Mod 1
SRS: Display Services	Available	Documentation	Andy Li	MK 2 Mod 1
SRS: Human Machine Interface	Available	Documentation	Andy Li	MK 2 Mod 1
SRS: Infrastructure Services (IS)	Available	Documentation	Andy Li	MK 2 Mod 1
SRS: Tactical Operations (TO)	Available	Documentation	Andy Li	MK 2 Mod 1
S/W: Tactical Operations Function	Available	Application	Andy Li	MK 2 Mod 1
S/W: OL	Available	Application	Andy Li	MK 2 MOD 1

2003 - 2008 Sponsored Research Topics

Acquisition Management

- Software Requirements for OA
- Managing Services Supply Chain
- Acquiring Combat Capability via Public-Private Partnerships (PPPs)
- Knowledge Value Added (KVA) + Real Options (RO) Applied to Shipyard Planning Processes
- Portfolio Optimization via KVA + RO
- MOSA Contracting Implications
- Strategy for Defense Acquisition Research
- Spiral Development
- BCA: Contractor vs. Organic Growth

Contract Management

- USAF IT Commodity Council
- Contractors in 21st Century Combat Zone
- Joint Contingency Contracting
- Navy Contract Writing Guide
- Commodity Sourcing Strategies
- Past Performance in Source Selection
- USMC Contingency Contracting
- Transforming DoD Contract Closeout
- Model for Optimizing Contingency Contracting Planning and Execution

Financial Management

- PPPs and Government Financing
- Energy Saving Contracts/DoD Mobile Assets
- Capital Budgeting for DoD
- Financing DoD Budget via PPPs
- ROI of Information Warfare Systems
- Acquisitions via leasing: MPS case
- Special Termination Liability in MDAPs

Human Resources

- Learning Management Systems
- Tuition Assistance
- Retention
- Indefinite Reenlistment
- Individual Augmentation

Logistics Management

- R-TOC Aegis Microwave Power Tubes
- Privatization-NOSL/NAWCI
- Army LOG MOD
- PBL (4)
- Contractors Supporting Military Operations
- RFID (4)
- Strategic Sourcing
- ASDS Product Support Analysis
- Analysis of LAV Depot Maintenance
- Diffusion/Variability on Vendor Performance Evaluation
- Optimizing CIWS Lifecycle Support (LCS)

Program Management

- Building Collaborative Capacity
- Knowledge, Responsibilities and Decision Rights in MDAPs
- KVA Applied to Aegis and SSDS
- Business Process Reengineering (BPR) for LCS Mission Module Acquisition
- Terminating Your Own Program
- Collaborative IT Tools Leveraging Competence

A complete listing and electronic copies of published research are available on our website: www.acquisitionresearch.org

Initial Distribution List

1. Defense Technical Information Center 2
   8725 John J. Kingman Rd., STE 0944; Ft. Belvoir, VA 22060-6218

2. Dudley Knox Library, Code 013 2
   Naval Postgraduate School, Monterey, CA 93943-5100

3. Research Office, Code 09 1
   Naval Postgraduate School, Monterey, CA 93943-5138

4. William R. Gates 1
   Dean, GSBPP
   Naval Postgraduate School, Monterey, CA 93943-5138

5. Stephen Mehay 1
   Associate Dean for Research, GB
   Naval Postgraduate School, Monterey, CA 93943-5138

6. Jean Johnson 1
   Research Assistant, SE
   Naval Postgraduate School, Monterey, CA 93943-5138

Copies of the Acquisition Sponsored Research Reports may be printed from our website: www.acquisitionresearch.org